

10-06-00

A

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)*(Only for new nonprovisional applications under 37 CFR 1.53(b))*Docket No.
13768.170Total Pages in this Submission
39**TO THE ASSISTANT COMMISSIONER FOR PATENTS**Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

SELECTING A SERVER TO SERVICE CLIENT REQUESTS

and invented by:

Sean Lyndersay
Brian Deen
Alex Hopmann
Joel SoderbergIf a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 31 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☒ Cross References to Related Applications *(if applicable)*
 - c. ☒ Statement Regarding Federally-sponsored Research/Development *(if applicable)*
 - d. ☒ Reference to Microfiche Appendix *(if applicable)*
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings *(if drawings filed)*
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.170

Total Pages in this Submission
39

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*
- a. ☒ Formal Number of Sheets 3
- b. ☐ Informal Number of Sheets _____
4. ☐ Oath or Declaration
- a. ☐ Newly executed *(original or copy)* ☐ Unexecuted
- b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
- c. ☐ With Power of Attorney ☐ Without Power of Attorney
- d. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under
Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby
incorporated by reference therein.
6. ☐ Computer Program in Microfiche *(Appendix)*
7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*
- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy *(identical to computer copy)*
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☐ Assignment Papers *(cover sheet & document(s))*
9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Acknowledgment postcard
14. ☒ Certificate of Mailing
- ☐ First Class ☒ Express Mail *(Specify Label No.):* EL695575009US

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.170

Total Pages in this Submission
39

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☒ Additional Enclosures *(please identify below):*

Form PTO-2038 submitting payment in the amount of \$1,042
Attachment regarding correspondence

Request That Application Not Be Published Pursuant To 35 U.S.C. 122(b)(2)

17. ☐ Pursuant to 35 U.S.C. 122(b)(2), Applicant hereby requests that this patent application not be published pursuant to 35 U.S.C. 122(b)(1). Applicant hereby certifies that the invention disclosed in this application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication of applications 18 months after filing of the application.

Warning

An applicant who makes a request not to publish, but who subsequently files in a foreign country or under a multilateral international agreement specified in 35 U.S.C. 122(b)(2)(B)(i), must notify the Director of such filing not later than 45 days after the date of the filing of such foreign or international application. A failure of the applicant to provide such notice within the prescribed period shall result in the application being regarded as abandoned, unless it is shown to the satisfaction of the Director that the delay in submitting the notice was unintentional.

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.170

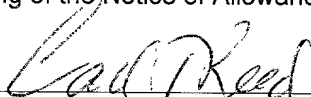
Total Pages in this Submission
39

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	34	- 20 =	14	x \$18.00	\$252.00
Indep. Claims	4	- 3 =	1	x \$80.00	\$80.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$710.00
OTHER FEE (specify purpose) _____					\$0.00
TOTAL FILING FEE					\$1,042.00

- ☒ A check in the amount of _____ to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **23-3178** as described below. A duplicate copy of this sheet is enclosed.
- ☐ Charge the amount of _____ as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).


Signature

Carl T. Reed
Registration No. 45,454

Dated: October 4, 2000



022913

CC:

PATENT TRADEMARK OFFICE

[illegible]

RICK D. NYDEGGER
WORKMAN, NYDEGGER & SEELEY
1000 Eagle Gate Tower
60 East South Temple
Salt Lake City, Utah 84111
Telephone: (801) 533-9800
Facsimile: (801) 328-1707

JC913 U.S. PTO
09/680120



CERTIFICATE OF MAILING BY "EXPRESS MAIL" (37 CFR 1.10)			Docket No. 13768.170
Applicant(s): Sean Lyndersay et al.			
Serial No.	Filing Date October 4, 2000	Examiner	Group Art Unit
Invention. SELECTING A SERVER TO SERVICE CLIENT REQUESTS			

I hereby certify that the following documents are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 in an envelope addressed to: The Assistant Commissioner for Patents, Washington, DC 20231.

Dated this 4th day of October 2000.

- Transmittal letter (4 pages) (in duplicate)
- Attachment for correspondence
- New patent application (31 pages)
- 3 sheets of drawings
- Form PTO-2038 submitting Credit Card Payment in the amount of \$1,042.00
- Postcard

Carl T. Reed, Registration No. 45,454

(Typed or Printed Name of Person Mailing Correspondence)


(Signature of Person Mailing Correspondence)

EL695575009US

("Express Mail" Mailing Label Number)

004001" 02F08960

UNITED STATES PATENT APPLICATION

of

Sean Lyndersay

Brian Deen

Alex Hopmann

and

Joel Soderberg

for

SELECTING A SERVER TO SERVICE CLIENT REQUESTS

2025-2026

004007" 03T02960

1 These systems are becoming increasingly popular and in some systems, including
2 the Internet, there is a great demand for the information that can be stored and managed by
3 server computers. The demand is often significant and a single server computer is often
4 not capable of meeting the demand. To increase accessibility, the data being accessed is
5 often distributed over a bank of server computers. Distributing the data across a server
6 bank, however, presents problems with regard to accessing the data.

7 For instance, when a new server is added to a server bank, the data is often
8 redistributed over the servers. As a result, the URI of the data may change and clients may
9 not have their requests fulfilled in these types of situations. Personal mailboxes in
10 particular are subject to this problem because they are typically maintained on a single
11 server. When a new server is added to the system and the personal mailboxes are
12 reallocated across the serves, the URI will change.

13 In order to access this bank of server computers, these systems provide a routing
14 server that receives the client request. However, the routing server usually stores some
15 data and actually services part of the client's request, including Hyper Text Markup
16 Language (HTML) rendering. In addition, the routing server does not really forward the
17 request to one of the servers in the server bank. Rather, the routing server typically uses a
18 separate protocol to access the data in the bank of server computers. The data returned
19 from this server is incorporated into the response sent to the client from the routing server.
20 As a result, the applications of this server system are limited to the applications of the
21 routing server.

22 Other systems simply use the routing server to route client requests to one of the
23 servers in the server bank. These systems, however, are not able to analyze the requests or
24 perform other operations on these requests other than forward them to one of the servers.

1 Necessarily, each server in the bank of servers completely replicates the data stored on the
2 other servers. This type of system makes no allowance for customized or selective
3 replication. For example, the routing server may be configured to route requests from a
4 particular geography to a particular server in the server bank. Even though this data may
5 be dedicated to a local topic that is of little interest to other users, this data must be
6 replicated on all of the servers in the server bank. The routing server operates under the
7 assumption that all of the data is available from each server. While this type of system
8 does provide a type of load balancing with regard to client requests, this system is unable
9 to respond dynamically at the time of the client request. These systems are also not
10 capable of immediately adapting to new situations, such as adding or removing servers
11 from the bank of servers. In addition, these types of systems cannot utilize selective data
12 replication across the servers.

13 A system that provides load balancing of client requests against multiple servers
14 therefore faces several issues that are difficult to overcome. For example, there is a need to
15 improve the server efficiency by servicing requests as quickly as possible. Additionally,
16 there is a need to allow access to the requested resources for as long as possible. Often,
17 client requests fail when the server selected by the routing server is unavailable. Finally,
18 there is a need to balance the requests across the available servers while dynamically
19 responding to both changing system conditions and multiple client requests.

SUMMARY OF THE INVENTION

The present invention relates to systems and methods for selecting a server for servicing client requests. The present invention includes a server system that has one or more back end servers connected with at least one front end server. The front end server receives client requests and directs those requests to one of the back end servers. The front end server is also capable of analyzing the client request such that the client requests are balanced across the back end servers. The present invention also balances the needs of timely servicing the client requests, distributing the client requests to the back end servers, allowing access to the requested data for as long as possible, and selecting a back end server that can actually service the client requests.

The selection of a back end server is often based on the user that is making the request, the resource that is the subject of the client request and the URI of the request. The user information can include the user name, the user's authentication token, and the like. The resource can be located in a private folder, a home public folder, or an application public folder. When a request is made to the private folders, the location of the associated user's private folder is determined by the front end server and the client request is forwarded to that server.

When a request is made to a default or home public folder, the user is directed to the server storing the home public folder that is associated with the private folder of the authenticated user. Identifying and selecting a server in this manner maintains the usefulness of status information related to the home public folder such as which items have been read and the like. In other words, client requests for private folders or the home public folder are usually sent to the same server. If the private folder is unavailable, the request is failed. If the home public folder is unavailable, the client request is redirected to

1 a server that replicates the home public folder. This ensures that a user has access to the
2 requested resource if it is available, while maintaining system efficiency.

3 When a request is made to an application public folder, the user's authentication
4 token is used as a key in a selection module that selects a back end server to service the
5 request. The selection module keeps a list of all servers such that the same server is
6 usually selected for a particular user. One example of the selection module is a hash
7 function that uses the user's authentication token as its key. If the selected back end server
8 is unavailable, then that server is marked as unavailable and the selection module is used to
9 identify an alternative back end server.

10 In some cases, a server does not store the actual content of a requested resource. In
11 that case, the server returns a list of back end servers to the front end server that do store
12 the requested content. The selection module operates on this shortened list to select a back
13 end server to service the client request.

14 Additional features and advantages of the invention will be set forth in the
15 description which follows, and in part will be obvious from the description, or may be
16 learned by the practice of the invention. The features and advantages of the invention may
17 be realized and obtained by means of the instruments and combinations particularly
18 pointed out in the appended claims. These and other features of the present invention will
19 become more fully apparent from the following description and appended claims, or may
20 be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an exemplary system that provides a suitable operating environment for the present invention;

Figure 2 is a block diagram illustrating an exemplary environment in which the systems and methods of the present invention may be implemented and includes a client connected to a server system that includes a front end server and one or more back end servers;

Figure 3 is a detailed block diagram of back end servers and illustrates the content stored on those servers; and

Figure 4 is a block diagram illustrating a front end server that selects a back end server to service a client request and directs the request to the selected server.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to systems and methods for using a front end server to direct a client request to a back end server. Client requests can generally be described as requests for private resources or folders and as requests to public resources or folders. In either case, the availability and capability of the back end servers should be accounted for when directing the client request.

While the present invention is described in terms of Hyper Text Transfer Protocol (HTTP), it is understood that the present invention can apply to other protocols as well such as Post Office Protocol 3 (POP3) and Internet Messaging Access Protocol (IMAP). When an HTTP request is initially received, the user may be prompted for authentication credentials. The authentication credentials may be used to select a back end server to service the client request. The front end server also utilizes the Uniform Resource Identifier (URI) of the client request to make a decision regarding which back end server will receive and service the client request. More generally, the URI is the primary determinant used by the front end server to select a back end server and the authentication information is often a secondary determinant. In cases where authentication credentials are not available, only the URI is used. In other cases, a user's authentication token is a primary determinant used by the front end server to select a back end server.

The advantage of using an intelligent front end server to direct requests is that the client requests will be properly distributed across all of the available back end servers. In addition, the front end server provides clients with prolonged accessibility, server efficiency, and directs client requests to back end servers that can actually service the client's request.

1 The present invention extends to both methods and systems for directing client
2 requests to back end servers. The embodiments of the present invention may comprise a
3 special purpose or general purpose computer including various computer hardware, as
4 discussed in greater detail below.

5 Embodiments within the scope of the present invention also include computer-
6 readable media for carrying or having computer-executable instructions or data structures
7 stored thereon. Such computer-readable media can be any available media which can be
8 accessed by a general purpose or special purpose computer. By way of example, and not
9 limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM
10 or other optical disk storage, magnetic disk storage or other magnetic storage devices, or
11 any other medium which can be used to carry or store desired program code means in the
12 form of computer-executable instructions or data structures and which can be accessed by
13 a general purpose or special purpose computer. When information is transferred or
14 provided over a network or another communications connection (either hardwired,
15 wireless, or a combination of hardwired or wireless) to a computer, the computer properly
16 views the connection as a computer-readable medium. Thus, any such a connection is
17 properly termed a computer-readable medium. Combinations of the above should also be
18 included within the scope of computer-readable media. Computer-executable instructions
19 comprise, for example, instructions and data which cause a general purpose computer,
20 special purpose computer, or special purpose processing device to perform a certain
21 function or group of functions.

22 Figure 1 and the following discussion are intended to provide a brief, general
23 description of a suitable computing environment in which the invention may be
24 implemented. Although not required, the invention will be described in the general context

1 of computer-executable instructions, such as program modules, being executed by
2 computers in network environments. Generally, program modules include routines,
3 programs, objects, components, data structures, etc. that perform particular tasks or
4 implement particular abstract data types. Computer-executable instructions, associated
5 data structures, and program modules represent examples of the program code means for
6 executing steps of the methods disclosed herein. The particular sequence of such
7 executable instructions or associated data structures represent examples of corresponding
8 acts for implementing the functions described in such steps.

9 Those skilled in the art will appreciate that the invention may be practiced in
10 network computing environments with many types of computer system configurations,
11 including personal computers, hand-held devices, multi-processor systems,
12 microprocessor-based or programmable consumer electronics, network PCs,
13 minicomputers, mainframe computers, and the like. The invention may also be practiced
14 in distributed computing environments where tasks are performed by local and remote
15 processing devices that are linked (either by hardwired links, wireless links, or by a
16 combination of hardwired or wireless links) through a communications network. In a
17 distributed computing environment, program modules may be located in both local and
18 remote memory storage devices.

19 With reference to Figure 1, an exemplary system for implementing the invention
20 includes a general purpose computing device in the form of a conventional computer 20,
21 including a processing unit 21, a system memory 22, and a system bus 23 that couples
22 various system components including the system memory 22 to the processing unit 21.
23 The system bus 23 may be any of several types of bus structures including a memory bus
24 or memory controller, a peripheral bus, and a local bus using any of a variety of bus

1 architectures. The system memory includes read only memory (ROM) 24 and random
2 access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic
3 routines that help transfer information between elements within the computer 20, such as
4 during start-up, may be stored in ROM 24.

5 The computer 20 may also include a magnetic hard disk drive 27 for reading from
6 and writing to a magnetic hard disk 39, a magnetic disk drive 28 for reading from or
7 writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or
8 writing to removable optical disk 31 such as a CD-ROM or other optical media. The
9 magnetic hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are
10 connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive-
11 interface 33, and an optical drive interface 34, respectively. The drives and their
12 associated computer-readable media provide nonvolatile storage of computer-executable
13 instructions, data structures, program modules and other data for the computer 20.
14 Although the exemplary environment described herein employs a magnetic hard disk 39, a
15 removable magnetic disk 29 and a removable optical disk 31, other types of computer
16 readable media for storing data can be used, including magnetic cassettes, flash memory
17 cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like.

18 Program code means comprising one or more program modules may be stored on
19 the hard disk 39, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an
20 operating system 35, one or more application programs 36, other program modules 37, and
21 program data 38. A user may enter commands and information into the computer 20
22 through keyboard 40, pointing device 42, or other input devices (not shown), such as a
23 microphone, joy stick, game pad, satellite dish, scanner, or the like. These and other input
24 devices are often connected to the processing unit 21 through a serial port interface 46

1 coupled to system bus 23. Alternatively, the input devices may be connected by other
2 interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor
3 47 or another display device is also connected to system bus 23 via an interface, such as
4 video adapter 48. In addition to the monitor, personal computers typically include other
5 peripheral output devices (not shown), such as speakers and printers.

6 The computer 20 may operate in a networked environment using logical
7 connections to one or more remote computers, such as remote computers 49a and 49b.
8 Remote computers 49a and 49b may each be another personal computer, a server, a router,
9 a network PC, a peer device or other common network node, and typically include many or
10 all of the elements described above relative to the computer 20, although only memory
11 storage devices 50a and 50b and their associated application programs 36a and 36b have
12 been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local
13 area network (LAN) 51 and a wide area network (WAN) 52 that are presented here by way
14 of example and not limitation. Such networking environments are commonplace in office-
15 wide or enterprise-wide computer networks, intranets and the Internet.

16 When used in a LAN networking environment, the computer 20 is connected to the
17 local network 51 through a network interface or adapter 53. When used in a WAN
18 networking environment, the computer 20 may include a modem 54, a wireless link, or
19 other means for establishing communications over the wide area network 52, such as the
20 Internet. The modem 54, which may be internal or external, is connected to the system bus
21 23 via the serial port interface 46. In a networked environment, program modules depicted
22 relative to the computer 20, or portions thereof, may be stored in the remote memory
23 storage device. It will be appreciated that the network connections shown are exemplary
24 and other means of establishing communications over wide area network 52 may be used.

1 Figure 2 is a block diagram that illustrates an environment in which the systems
2 and methods of the present invention may be implemented. In Figure 2, a client 200 is
3 connected to a server system 201. The server system 201 includes a front end server 210
4 and one or more back end servers 220. The front end server 210 in this example is
5 stateless and does not have or store any content or data that is the subject of the client
6 request. The front end server 210 also does not require, for example, a hard disk or other
7 form of memory for storing program data. However, the front end server 210 is not
8 precluded from storing or maintaining data. The front end server 210, however, may store
9 data used to select an appropriate back end server to service the client request.

10 Primarily, the front end server 210 receives requests from the client 200 and directs
11 those requests to one of the back end servers 220. The front end server 210 thus provides a
12 single namespace from the viewpoint of the client 200 and the client 200 is unaware of
13 which back end server actually services the request. In this sense, the front end server 210
14 functions to direct or route client requests to the appropriate server. This is more fully
15 described in a co-pending application entitled "Routing Client Requests to Back-End
16 Servers," serial number __/_____, filed the same day herewith and incorporated by
17 reference in its entirety.

18 Figure 3 is a block diagram that more fully illustrates the back end servers 220.
19 The folder hierarchy or structure of each server included in the back end servers 220 is also
20 shown. The back end servers 220 includes, in this example, server 222 and server 224.
21 The server 222 stores private folders 226 and public folders 230, and the server 224 stores
22 the private folders 228 and the public folders 232. The private folders 226 and 228 may be
23 used, for example, as user mailboxes. As a result, the private folders 226 and 228 may
24 have a similar structure, but the content of the private folders 226 and 228 is different

2025 OCT 07 09:00

1 because they correspond to different clients or users. More specifically, a particular user's
2 mailbox is only located on a single back end server. For that reason, the mailboxes or
3 content stored in the private folders 226 are not replicated in the private folders 228. Users
4 are typically assigned to a particular server, which has its own private folders. When a
5 user makes a request for a private folder, the request is directed to a specific server that
6 stores the user's private content.

7 On the other hand, the public folders 230 and 232 not only have similar structure or
8 hierarchy, but also have the same content in most instances. In structure, the folder 233 is
9 identical to the folder 238, the folder 234 is identical to the folder 239, the folder 236 is
10 identical to the folder 240, the folder 235 is identical to the folder 241, and the folder 237
11 is identical to the folder 242.

12 In content, however, the public folders 230 may differ from the public folders 232.
13 The difference in content occurs because it is not always necessary for each back end
14 server to maintain all of the content that may be found on other back end servers. For
15 example, the folder 236 and the folder 240 are corresponding folders in identical folder
16 trees, but are located on different back end servers. The folder 236, which is illustrated in
17 a dashed line, does not contain any content and is referred to herein as a ghosted folder.
18 The folder 240, which is illustrated in a solid line, does contain content. If necessary, the
19 content of the folder 240 can be replicated to the folder 236 and the folder 236 would no
20 longer be ghosted.

21 One of the reasons for organizing the public folders in this manner is related to how
22 often a particular folder is accessed. Where content is actually stored is also related to
23 factors such as geography that may determine which back end server is selected. For
24 example, the private folders of users in a particular geographical location may all be stored

1 on a single server. The home public folder that is associated with those users may contain
2 content that is specific to those users and of little interest to others. As a result, the content
3 of that folder does not need to be replicated on other back end servers because those users
4 are usually directed to their home public folder. Thus, the content may be ghosted on the
5 other back-end servers and replicated as needed.

6 Another aspect of the back end servers 220 is that each private folder is usually
7 associated with a specific public folder. In Figure 3, the private folders 226 are associated
8 with the public folders 230 while the private folders 228 are associated with the public
9 folders 232. From the viewpoint of the private folders 226, the public folders 230 are
10 referred to as home public folders, while the public folders 232 are referred to as
11 application public folders. A user is preferably directed to their home public folder when
12 requesting content from the public folders. If the home public folder is unavailable or
13 contains a ghosted folder, then the user may be directed to an application public folder that
14 actually contains the requested data. The back end servers 220 are capable of functioning
15 with application public folders as well as other configurations of top level hierarchies or
16 data configurations.

17 The front end server, shown in Figure 2, is primarily responsible for directing client
18 requests to one of the back end servers. The selection of a back end server to service client
19 requests is a difficult task and several factors, such as server efficiency, system efficiency,
20 accessibility, and correctness, should be considered. Server efficiency relates to servicing
21 the client request as quickly as possible. System efficiency relates to balancing all of the
22 client requests across all of the available back end servers such that each of the back end
23 servers does not service an inordinate amount of client requests. Accessibility refers to
24 affording a client with access to a particular resource for as long as possible and

1 correctness refers to selecting a back end server that can actually service the client's
2 request.

3 When a back end server is selected or accessed, client requests to private folders
4 and client requests to public folders should be considered. In each case, the client request
5 can be affected by whether the selected back end server is available and whether the back
6 end server can actually service the request. Availability may be a temporary condition
7 while the ability of the back end server to service the request may be dependent on
8 compatibility issues and is a more permanent condition.

9 Figure 4 is a block diagram that illustrates how a back end server is selected by the
10 front end server to service a client request. The front end server 210 has access to a
11 directory 211 that contains metadata including configuration information. The
12 configuration information includes, but is not limited to, mapping users to their mailbox
13 server and their mailbox store, mapping a mailbox store to a public folder server, and
14 mapping a public folder tree to public folder servers.

15 The front end server 210 either accesses the directory 211 directly to obtain the
16 configuration information, or maintains a copy of the directory 211. In the case where the
17 private folders represent personal mailboxes, the directory 211 may be used to identify
18 which mailbox server and which mailbox store are associated with a particular user. The
19 directory 211 also identifies the home public folder that is associated with a particular
20 mailbox store. In this manner, the home public folder for a particular user may be
21 determined. The directory 211 may also be accessed to determine which servers maintain
22 a copy of a particular public folder tree. It is possible that more than one public folder tree
23 can be present on the back end servers. Additionally, all of the servers are not required to
24

1 have all of the public folder trees. More generally, the data stored on the back-end servers
2 does not have to be replicated in its entirety on all of the back end servers.

3 When a request is received from a client 200, an initial determination is made by
4 the front end server 210 as to whether the client request is directed to the user's private
5 folders, the home public folder, or to an application public folder. Using the directory 211,
6 the front end server 210 examines the Uniform Resource Identifier (URI) received in the
7 client request to identify the resource the user is attempting to access (read, write, copy,
8 move, save, delete and the like), and whether the client request is directed to the private
9 folders, the home public folders, or to application public folders. If the URI cannot be
10 identified as one of these types, then the client request is typically failed.

11 For example, if a client request contains a URI of "http://server/mailbox/user," then
12 this request is intended for the mailbox of the user, which is located in the private folders.
13 The front end server 210, in this case, determines which back end server is the mailbox
14 server for this user from the directory 211, which contains metadata identifying the
15 location of the user's mailbox. If the URI of the client request is http://server/mailbox,
16 then the front end server 210 will look up the username associated with the authenticated
17 user and proceed as if the request had contained the username of the authenticated user, as
18 described above. After the back end server is determined or selected, the client request is
19 directed to the identified back end server.

20 Another situation handled by the front end server 210 is where the client is
21 attempting to access the public folders. When the client request is received at the front end
22 server 210, the configuration information in or from the directory 211 is used to determine
23 that the client request is directed to the public folders. Within the directory 211, the virtual
24 root associated with the URI of the client request is mapped to a physical location. The

1 path name of the actual location of the requested data may contain the string “public
2 folder.” If this string is detected in the path, then the front end server 210 determines that
3 the client request is directed to the public folder. Even though the virtual root may change,
4 the actual physical location will most likely contain a string that can be used to determine
5 that the client request is directed to the public folders. The actual string used to make this
6 determination can vary and is not limited to “public folder.” It is understood that the front
7 end server 210 is not precluded from determining that the client request is directed to the
8 public folders using other methods.

9 The next issue is determining which back end server will serve this client request to
10 access the public folders. As previously described, each user has an associated home
11 public folder. The home public folder of a particular user is associated with the private
12 folder of that user and is typically located on the same back end server as the private
13 folders of the user. Thus, a request for the public folder is typically sent to the home
14 public folder every time a new request from a particular user is received by the front end
15 server 210.

16 This is important because each public folder on each back end server is effectively
17 the same as described with reference to figure 3. However, only the home public folder of
18 a user stores status information such as when a user has read an item and which items have
19 not been read. Maintaining this type of information is particularly useful when a user is
20 always directed to the same home public folder. This information would not be useful if a
21 user were not directed to the same back end server. Thus, the front end server 210 directs
22 all requests from a particular user to that user’s home public folders when possible.

23 The next situation involves a URI that refers to an application public folder or top
24 level hierarchy system. The front end server 210 obtains a list of servers 212 from the

1 directory 211 that have a replica of the application folder tree that is referenced by the URI
2 in the client request. The front end server 210 also has access to the user's authentication
3 token or security identifier (SID) 214. In order to determine which back end server to send
4 the client request to, the front end server 210 uses the SID 214 as input to the selection
5 module 215 in order to select one of the back end servers in the list of servers 212. One
6 example of the selection module 215 is a hash function.

7 Using the SID 214 as a key in a hash function ensures that exactly one server in the
8 list of servers 212 will be selected and that the same server will be selected for the user
9 assuming that the number of back end servers does not change. In this manner, client
10 requests from a particular user are sent to the same server when possible. In addition, the
11 hash function is able to spread the client requests across all of the available back end
12 servers 220 because the SID 214 of each user is different. Thus, the system efficiency is
13 maintained by distributing the client requests across all of the back end servers 220.

14 As shown in Figure 4, after a back end server has been selected, the request 216 is
15 forwarded to the selected server, which is shown as back end server 222. The back end
16 server 222 services the request 216 if possible. If the back end server 222 is unavailable,
17 then the server is marked as unavailable in the list of servers 212. For illustrative
18 purposes, the list of servers 212 with servers marked as unavailable is shown as the list of
19 servers 213.

20 When a selected back end server is unavailable, nothing is usually returned to the
21 front end server 210, although that server is marked as unavailable as described. When the
22 selected back end server is unavailable, the hash function is performed against the
23 remaining servers in the list of servers 213 in order to select a different back end server to
24 service the client request.

1 In some instances, the front end server 210 will already have some of the servers in
2 the list of servers 212 marked as unavailable when an initial client request is received.
3 However, these servers are not excluded from the hash function in order to ensure that the
4 same back end server is repeatedly selected. Altering the number of servers in the list of
5 servers 212 could result in an uneven distribution of client requests across the back end
6 servers 220. If, however, the hash function does select a server that is marked as
7 unavailable, then that server is removed from the list of servers 212 for that request only
8 and the hash is re-executed to select a back end server to service the request. This process
9 can be repeated until either a functioning back end server is selected or it is determined that
10 none of the back end servers host the requested content. Servers that are marked as
11 unavailable are periodically reevaluated to determine their availability and the mark is
12 removed when they are available.

13 Once a valid back end server has been selected, the request 216 is sent to the back
14 end server 222 to service the request. In this case, the back end server 222 has resource
15 240 and resource 241. The resource 241, indicated by dashed lines, is ghosted and the
16 content of that resource is not physically present on the back end server 222. If the request
17 216 is for the ghosted resource 241, the back end server 222 sends a list of servers 218 to
18 the front end server 210 that contains a list of all back end servers that actually do store the
19 requested content. The list of servers 218 returned by the back end server 222 is typically
20 a subset of the list of servers 212. The front end server 210 executes the hash function on
21 the list of servers 218 to select a back end server to service the client request. In this
22 example, the back end server 224 is selected and the request 217 is then sent to the back
23 end server 224, which does store the content in the resource 243 that is ghosted in the
24 resource 241 and was the subject of the initial request 216. As more fully described in the

1 co-pending U.S. application serial No. ___/___ entitled "Transparently Redirecting
2 Client Requests for Content," filed the same day herewith and incorporated by reference,
3 the redirection of the request 216 from the back end server 222 to the back end server 224
4 is transparent.

5 Another aspect of the present invention is that when the front end server 210 is
6 generating the initial list of servers 212, one or more tiers of servers are generated. For
7 example, the first tier of servers may include back end servers that provide fast access
8 when compared to other back end servers. Alternatively, some back end servers included
9 in the first tier may have larger bandwidths than other back end servers. These types of
10 back end servers are included in the first tier of servers in the list of servers 212, while the
11 remaining servers are classified into other tiers. The front end server 210 usually attempts
12 to select a back end server that is in the first tier when selecting a back end server to
13 service a client request. If a first tier server is not selected, the systems and methods
14 described herein will be applied to the second tier of servers.

15 The present invention may be embodied in other specific forms without departing
16 from its spirit or essential characteristics. The described embodiments are to be considered
17 in all respects only as illustrative and not restrictive. The scope of the invention is,
18 therefore, indicated by the appended claims rather than by the foregoing description. All
19 changes which come within the meaning and range of equivalency of the claims are to be
20 embraced within their scope.

21 What is claimed and desired to be secured by United States Letters Patent is:
22
23
24

1. In a system including a client and a server system, the server system including a front end server and one or more back end servers, each back end server having private folders and public folders, wherein a client request is received at the front end server, a method for selecting a particular back end server to service the client request, the method comprising the acts of:

accessing a directory for configuration information based on a Uniform Resource Identifier in the client request by the front end server; and

selecting the particular back end server to service the client request by the front end server, wherein the particular back end server is one of:

a first back end server identified from the configuration information if the client request is for a private folder, wherein the first back end server stores the private folder of the client;

a second back end server identified from the configuration information if the client request is for a home public folder, wherein the second back end server identified from the configuration information stores the home public folder associated with the private folder of the client; and

a third back end server selected with a selection module if the client request is for an application public folder, wherein the selection module identifies the third back end server for each client request of a particular user.

2. A method as defined in claim 1, further comprising the act of directing the client request to the particular back end server.

1 3. A method as defined in claim 1, further comprising the act of servicing the
2 client request with the particular back end server.

3
4 4. A method as defined in claim 1, wherein the configuration information
5 associates a username with the private folder, and wherein the username is identified from
6 one of the URI of the client request or authentication information of a user.

7
8 5. A method as defined in claim 1, wherein the configuration information
9 associates the home public folder with the private folder.

10
11 6. A method as defined in claim 1, wherein the selection module further
12 comprises a hash function, wherein a list of the one or more back end servers is hashed
13 with a security token of a user to select the particular back end server.

14
15 7. A method as defined in claim 6, wherein one or more of the back end
16 servers included in the list of the one or more back end servers is marked as unavailable,
17 wherein the hash function does not exclude a back end servers that is marked as
18 unavailable unless the particular back end server selected by the hash function is marked as
19 unavailable.

20
21 8. A method as defined in claim 1, wherein the back end servers include
22 ghosted folders.

23
24

1 9. A method as defined in claim 8, further comprising the act or returning a
2 subset of servers to the front end server if the particular back end server ghosts a content
3 requested in the client request.

4
5 10. A method as defined in claim 9, wherein the selection module selects a new
6 back end server from the subset of servers to service the client request.

7
8 11. A method as defined in claim 10, wherein the one or more back end servers
9 included in the subset of servers actually store the content requested in the client request.

10
11 12. A method as defined in claim 10, further comprising the act of directing the
12 client request to the new back end server.

13
14 14. A method as defined in claim 1, wherein the first back end server is the
15 same as the second back end server.

16
17 15. A computer readable medium having computer executable instructions for
18 performing the acts recited in claim 1.

1 16. In a system including a client and a server system including a front end
2 server and one or more back end servers, a method for directing a client request to a
3 particular back end server, the particular back end server included in the one or more back
4 end servers, the method comprising the acts of:

5 identifying the particular back end server from a directory if the client
6 request is for a private folder, wherein only the particular back end server stores the
7 private folder of the client;

8 identifying the particular back end server from the directory if the client
9 request is for a home public folder, wherein only the particular back end server
10 stores the home public folder of the client;

11 identifying the particular back end server from a selection module if the
12 particular back end server cannot be identified from the directory, wherein the
13 selection module identifies the same particular back end server for each client
14 request of a particular user; and

15 directing the client request to the particular back end server.

16
17 17. A method as defined in claim 16, wherein the directory associates a URI of
18 the client request with the private folder, associates the private folder with the home public
19 folder, and associates each of the one or more back end servers that have a replica of the
20 home public folder.

21
22 18. A method as defined in claim 16, wherein the act of identifying the
23 particular back end server from a directory if the client request is for a private folder
24 further comprises the act of examining the URI to determine a username.

004037-0200360

1
2 19. A method as defined in claim 16, wherein the act of identifying the
3 particular back end server from the directory if the client request is for a home public
4 folder further comprises the act of identifying a string in the directory to determine that the
5 client request is for a home public folder.

6
7 20. A method as defined in claim 19, wherein the string is "public folder."

8
9 21. A method as defined in claim 16, wherein the selection module performs a
10 hash function using a list of the one or more back end servers and a security token of a user
11 to select the particular back end server.

22. In a system including a client and a server system including a front end server and one or more back end servers, a method for directing a client request for content in an application public folder to a particular back end server, the method comprising the acts of:

identifying a security token of a user associated with the client request;

identifying a list of back end servers from a directory, wherein each back end server in the list of back end servers has a replica of the application public folder;

selecting the particular back end server from the list using a selection module, wherein the selection module hashes the security token of the user and the list of back end servers; and

directing the client request to the particular back end server selected by the selection module.

23. A method as defined in claim 22, wherein the particular back end server selected by the selection module ghosts the content requested in the client request.

24. A method as defined in claim 23, further comprising the acts of:

the particular back end server sending a subset list of back end servers to the front end server, wherein the subset list includes each of the one or more back end servers that actually stores the content;

selecting a new back end server from the subset list with the selection module; and

directing the client request to the new back end server.

1
2 25. A method as defined in claim 22, wherein the particular back end server is
3 unavailable.

4
5 26. A method as defined in claim 25, further comprising the acts of:
6 marking the particular server as unavailable in the list of back end servers;
7 selecting a new particular server from the list of back end servers with the
8 selection module, wherein the unavailable back end server is not excluded from the
9 list of back end servers unless the unavailable back end server is selected by the
10 selection module; and
11 directing the client request to the new particular server.

12
13 27. A method as defined in claim 22, wherein at least one of the back end
14 servers included in the list of back end servers is marked as unavailable and wherein the
15 act of selecting the particular back end server comprises the act of keeping the marked
16 back end servers in the list of back end servers when the selection module selects the
17 particular back end server.

18
19 28. A computer readable medium containing computer executable instructions
20 for executing the acts recited in claim 22.
21
22
23
24

29. In a system including a client and a server system, the server system including a front end server and one or more back end servers, a computer program product for implementing a method for directing a client request to a particular back end server, the computer program product comprising:

a computer-readable medium carrying executable instructions for performing the method, wherein the method includes the acts of:

identifying the particular back end server from a directory if the client request is for a private folder, wherein only one of the one or more back end servers stores the private folder of the client;

identifying the particular back end server from the directory if the client request is for a home public folder, wherein only one of the one or more back end servers stores the home public folder of the client;

identifying the particular back end server from a selection module if the particular back end server cannot be identified from the directory, wherein the selection module identifies the same particular back end server for each client request of a particular user; and

directing the client request to the particular back end server.

30. A computer program product as defined in claim 29, wherein the directory associates a URI of the client request with the private folder, associates the private folder with the home public folder, and associates each of the one or more back end servers that have a replica of the home public folder.

1 31. A computer program product as defined in claim 29, wherein the act of
2 identifying the particular back end server from a directory if the client request is for a
3 private folder further comprises the act of examining the URI to determine a username.
4

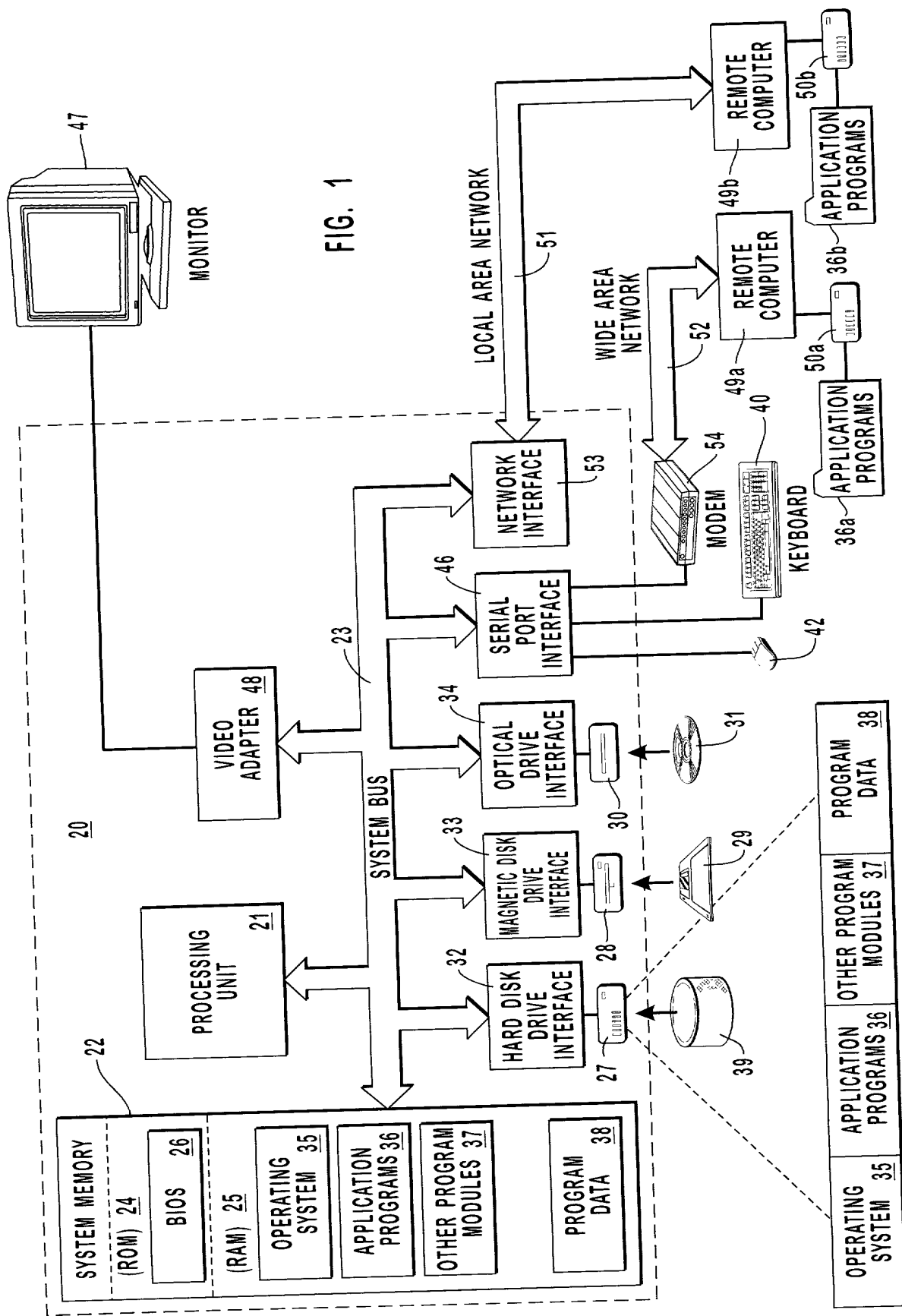
5 32. A computer program product as defined in claim 29, wherein the act of
6 identifying the particular back end server from the directory if the client request is for a
7 home public folder further comprises the act of identifying a string in the directory to
8 determine that the client request is for a home public folder.
9

10 33. A computer program product as defined in claim 29, wherein the string is
11 “public folder.”
12

13 34. A computer program product as defined in claim 29, wherein the selection
14 module performs a hash function using a list of the one or more back end servers and a
15 security token of a user to select the particular back end server.
16
17
18
19
20
21
22
23
24

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

G:\DATA\PAT\WORDPAT\13768 170.doc



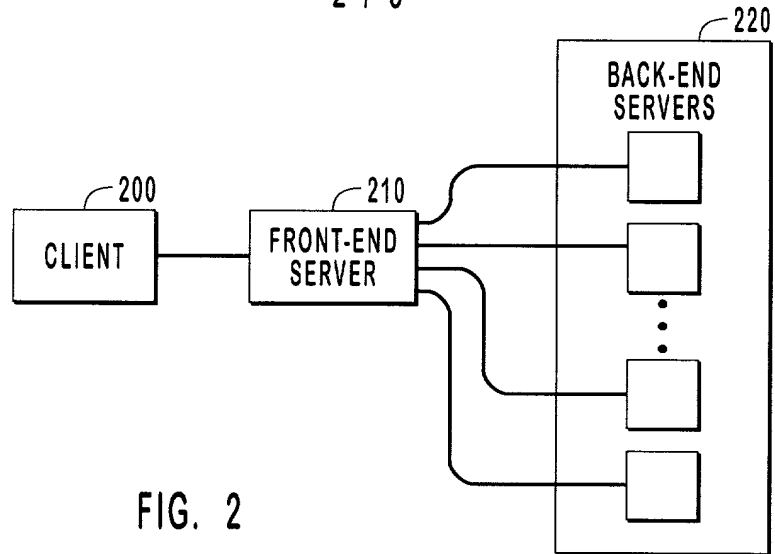


FIG. 2

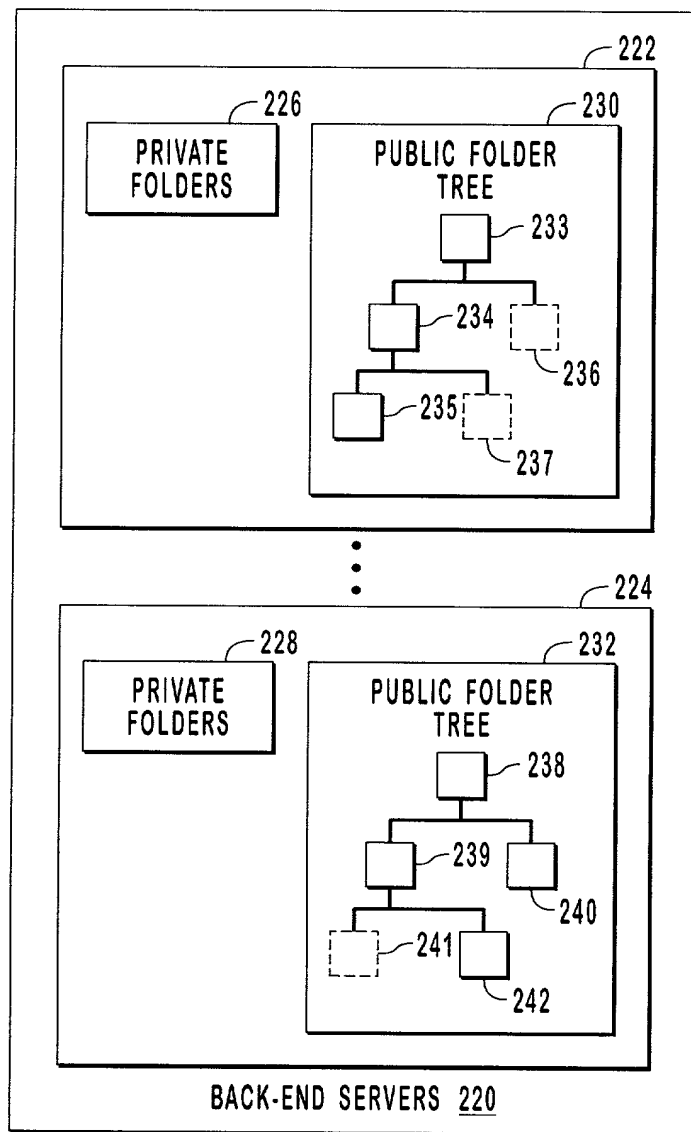


FIG. 3

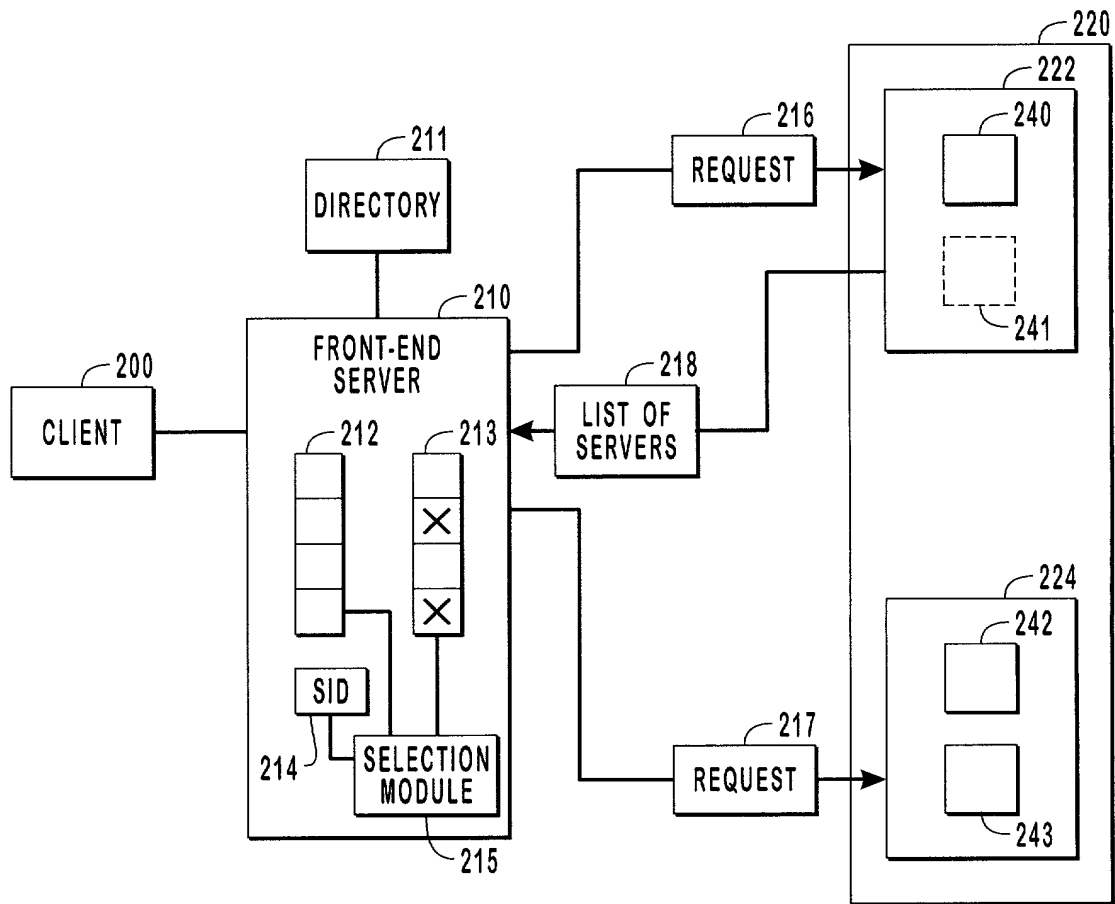


FIG. 4